JC20 Rec'd PCT/PTO 30 JUL 2001

FORM PTO-1390 (Modified) (REV 11-98) U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE

TRANSMITTAL LETTER TO THE UNITED STATES DESIGNATED/ELECTED OFFICE (DO/EO/US) CONCERNING A FILING UNDER 35 U.S.C. 371

ATTORNEY'S DOCKET NUMBER: **112740-271**

U.S. APPLICATION NO. (IF KNOWN, SEE 37 CFR: 09/890432

INTERNATIONAL APPLICATION NO.	INTERNATIONAL FILING DATE	PRIORITY DATE CLAIMED
PCT/DE00/00077	**11 January 2000**	**29 January 1999**

TITLE OF INVENTION

A METHOD FOR SECURING ACCESS TO AT LEAST ONE VARIABLE IN A PREEMPTIVELY MULTITASKING CONTROLLED PROCESSOR SYSTEM

APPLICANT(S) FOR DO/EO/US

Dr. Gerhard Spitz

Applicant herewith submits to the United States Designated/Elected Office (DO/EO/US) the following items and other information:

1. ☒ This is a **FIRST** submission of items concerning a filing under 35 U.S.C. 371.
2. ☐ This is a **SECOND** or **SUBSEQUENT** submission of items concerning a filing under 35 U.S.C. 371.
3. ☒ This is an express request to begin national examination procedures (35 U.S.C. 371(f)) at any time rather than delay examination until the expiration of the applicable time limit set in 35 U.S.C. 371(b) and PCT Articles 22 and 39(1).
4. ☒ A proper Demand for International Preliminary Examination was made by the 19th month from the earliest claimed priority date.
5. ☒ A copy of the International Application as filed (35 U.S.C. 371 (c) (2))
   a. ☒ is transmitted herewith (required only if not transmitted by the International Bureau).
   b. ☐ has been transmitted by the International Bureau.
   c. ☐ is not required, as the application was filed in the United States Receiving Office (RO/US).
6. ☒ A translation of the International Application into English (35 U.S.C. 371(c)(2)).
7. ☒ A copy of the International Search Report (PCT/ISA/210).
8. ☒ Amendments to the claims of the International Application under PCT Article 19 (35 U.S.C. 371 (c)(3))
   a. ☒ are transmitted herewith (required only if not transmitted by the International Bureau).
   b. ☐ have been transmitted by the International Bureau.
   c. ☐ have not been made; however, the time limit for making such amendments has NOT expired.
   d. ☐ have not been made and will not be made.
9. ☒ A translation of the amendments to the claims under PCT Article 19 (35 U.S.C. 371(c)(3)).
10. ☒ An oath or declaration of the inventor(s) (35 U.S.C. 371 (c)(4)).
11. ☒ A copy of the International Preliminary Examination Report (PCT/IPEA/409).
12. ☐ A translation of the annexes to the International Preliminary Examination Report under PCT Article 36 (35 U.S.C. 371 (c)(5)).

Items 13 to 20 below concern document(s) or information included:

13. ☐ An Information Disclosure Statement under 37 CFR 1.97 and 1.98.
14. ☒ An assignment document for recording. A separate cover sheet in compliance with 37 CFR 3.28 and 3.31 is included.
15. ☒ A **FIRST** preliminary amendment.
16. ☐ A **SECOND** or **SUBSEQUENT** preliminary amendment.
17. ☒ A substitute specification.
18. ☐ A change of power of attorney and/or address letter.
19. ☒ Certificate of Mailing by Express Mail
20. ☒ Other items or information:

Submission of Drawings Figure 1 on one sheet

PCTUS1/REV03

JC17 Rec'd PCT/PTO 30 JUL 2001

U.S. APPLICATION NO. (IF KNOWN, SEE 37 CFR	INTERNATIONAL APPLICATION NO.	ATTORNEY'S DOCKET NUMBER
09/890432	PCT/DE00/00077	112740-271

21. The following fees are submitted:.

BASIC NATIONAL FEE (37 CFR 1.492 (a) (1) - (5)) :

	CALCULATIONS	PTO USE ONLY
☐ Neither international preliminary examination fee (37 CFR 1.482) nor international search fee (37 CFR 1.445(a)(2) paid to USPTO and International Search Report not prepared by the EPO or JPO **$1,000.00**		
☒ International preliminary examination fee (37 CFR 1.482) not paid to USPTO but Internation Search Report prepared by the EPO or JPO **$860.00**		
☐ International preliminary examination fee (37 CFR 1.482) not paid to USPTO but international search fee (37 CFR 1.445(a)(2)) paid to USPTO **$710.00**		
☐ International preliminary examination fee paid to USPTO (37 CFR 1.482) but all claims did not satisfy provisions of PCT Article 33(1)-(4) **$690.00**		
☐ International preliminary examination fee paid to USPTO (37 CFR 1.482) and all claims satisfied provisions of PCT Article 33(1)-(4) **$100.00**		
ENTER APPROPRIATE BASIC FEE AMOUNT =	**$860.00**	
Surcharge of **$130.00** for furnishing the oath or declaration later than ☐ 20 ☐ 30 months from the earliest claimed priority date (37 CFR 1.492 (e)).	**$0.00**	

CLAIMS	NUMBER FILED	NUMBER EXTRA	RATE		
Total claims	6 - 20 =	0	x $18.00	**$0.00**	
Independent claims	1 - 3 =	0	x $80.00	**$0.00**	
Multiple Dependent Claims **(check if applicable)**.			☐	**$0.00**	
	TOTAL OF ABOVE CALCULATIONS		=	**$860.00**	
Reduction of 1/2 for filing by small entity, if applicable. Verified Small Entity Statement must also be filed (Note 37 CFR 1.9, 1.27, 1.28) **(check if applicable)**.			☐	**$0.00**	
	SUBTOTAL		=	**$860.00**	
Processing fee of **$130.00** for furnishing the English translation later than ☐ 20 ☐ 30 months from the earliest claimed priority date (37 CFR 1.492 (f)).			+	**$0.00**	
	TOTAL NATIONAL FEE		=	**$860.00**	
Fee for recording the enclosed assignment (37 CFR 1.21(h)). The assignment must be accompanied by an appropriate cover sheet (37 CFR 3.28, 3.31) **(check if applicable)**.			☐	**$0.00**	
	TOTAL FEES ENCLOSED		=	**$860.00**	
				Amount to be: refunded	$
				charged	$

☒ A check in the amount of **$860.00** to cover the above fees is enclosed.

☐ Please charge my Deposit Account No. in the amount of to cover the above fees. A duplicate copy of this sheet is enclosed.

☒ The Commissioner is hereby authorized to charge any fees which may be required, or credit any overpayment to Deposit Account No. **02-1818** A duplicate copy of this sheet is enclosed.

NOTE: Where an appropriate time limit under 37 CFR 1.494 or 1.495 has not been met, a petition to revive (37 CFR 1.137(a) or (b)) must be filed and granted to restore the application to pending status.

SEND ALL CORRESPONDENCE TO:

William E. Vaughan (Reg. No. 39,056)
Bell, Boyd & Lloyd LLC
P.O. Box 1135
Chicago, Illinois 60690

SIGNATURE

William E. Vaughan
NAME

39,056
REGISTRATION NUMBER

July 30, 2001
DATE

BOX PCT

IN THE UNITED STATES ELECTED/DESIGNATED OFFICE
OF THE UNITED STATES PATENT AND TRADEMARK OFFICE
UNDER THE PATENT COOPERATION TREATY-CHAPTER II

PRELIMINARY AMENDMENT

APPLICANT: Dr. Gerhard Spitz DOCKET NO: 112740-271
SERIAL NO: GROUP ART UNIT:
EXAMINER:
INTERNATIONAL APPLICATION NO: PCT/DE00/00077
INTERNATIONAL FILING DATE: 11 January 2000
INVENTION: A METHOD FOR SECURING ACCESS TO AT LEAST ONE VARIABLE IN A PREEMPTIVELY MULTITASKING-CONTROLLED PROCESSOR SYSTEM

Assistant Commissioner for Patents,
Washington, D.C. 20231

Sir:

Please amend the above-identified International Application before entry into the National stage before the U.S. Patent and Trademark Office under 35 U.S.C. §371 as follows:

In the Specification:

Please replace the Specification of the present application, including the Abstract, with the following Substitute Specification:

SPECIFICATION

TITLE

MULTITASKING-CONTROLLED PROCESSOR SYSTEM

BACKGROUND OF THE INVENTION

Field of the Invention

In existing and future information processing systems, such as personal computers, software objects (also referred to as processes) - are and will be administered using the operating system in such a way that the hardware system, in

particular the process-processing device which is provided in the information processing system, such as the processor, is utilized uniformly with the aim of high overall efficiency. In this way, the software modules which are assigned to the processor by the operating system (also referred to as tasks) are processed by the processor. Here, special operating systems, for example Windows 95, are provided for the information processing systems which have a monoprocessor, i.e. the information processing system has just one processor, the operating systems also permitting multi-user operation or multiple-process operation on a monoprocessor -- see in this respect "Architektur von Betriebssystemen" [Architecture of Operating Systems], H. Wetterstein, Hanser Studien Bücher [publishing house], 1984, pp. 54 et seq. The operating mode which is required for the multiple-process operation of a processor is known in the specialist field under the term "multiprogramming" or else "multitasking". In this way, during the execution of a task the information processing system can also carry out a further task such as the reading of data from a storage medium of the information processing system or, for example, the displaying of data on a data viewing station in a "quasiparallel" fashion.

Furthermore, a distinction is made between "cooperative" and "preemptive" multitasking. In the case of "cooperative" multitasking, each individual currently executed task itself determines, according to requirements, the time period for which it takes up the processor; i.e., the currently running task decides on the time when the processor is released for the processing of further tasks. In the case of "preemptive" multitasking, a task of the operating system, known in the specialist field as "scheduler" or even "task scheduler", interrupts the currently executed task after a predefined or assigned time period has finished; i.e., the time when the processor is assigned and released is determined using the task scheduler.

In order to execute a function of the operating system, for example an operating system task such as the task scheduler, a special operating mode of the processor for protecting the data of the operating system task is provided which is known as supervisor or kernel mode - see Andrew S. Tanenbaum, "Betriebssysteme - Entwurf und Realisierung" [Operating Systems - Design and Implementation] part 1, Prentice-Hall International, 1990, pp 31/32. To do this, the processor is switched over using a supervisor call from a user mode into the supervisor mode and the control of the

processor is thus transferred to the operating system or its tasks. In contrast with the supervisor mode, not all instructions are acceptable in the user mode, inter alia, in the user mode the use of input and output instructions and of some special instructions is prohibited. Likewise, in the user mode the access to all the data is generally not possible, for example the data of the operating system can neither be read nor amended for non-operating system tasks.

Specifically in the case of information processing systems which act according to the multitasking principle, variables or blocks of variables which are accessed during the processing of a task must be protected against competing accesses, for example by further tasks. This ensures that, for example, the errors occurring during dual simultaneous variable access cannot lead to any blockages of further tasks or of the entire information processing system. Such a protection mechanism is described below using the formulation "secured access" to at least one variable, and the term variable can refer here both to a variable of a software module which is stored in a memory unit and to a hardware-related setting information item which is stored in a hardware register. Such secured accesses frequently take place when specific problems are posed, for example in information systems which are used to control real time systems but must also access data which can be administrated, and are of short duration in comparison to the average time period between two successive task changes. Consequently, the probability of a task change during a secure access is very low, but cannot at all be excluded.

The implementation of a "secure access" by a task can be carried out using various protection mechanisms. This includes, inter alia, the setting of a task change inhibit in order to avoid a competing access by a further task to the variables which are being accessed by the task currently running on the processor. To do this, before the variables to be read are accessed using a supervisor call, the processor is switched over into the supervisor mode and the setting of a task change inhibit is requested from the operating system in order to obtain exclusive access for the processor, and thus also for the desired variable, for the currently accessing task. Then, the processor is switched back into the user mode and the desired access to the variable can be secured by the previously interrupted task; i.e., without interruption. After termination of the secure access by the currently running task, it is necessary to change again into the supervisor

mode via a supervisor call and for the task change inhibit to be reset by the operating system in the supervisor mode. In order to further process the task which is currently to be processed, the processor is then changed back into the user mode and the time monitoring activated during the setting of the task change inhibit is deactivated in order to avoid the processor being blocked for an indeterminately long time.

A further method of implementing a secure access is used in the synchronization of tasks, i.e. the coordination of a number of tasks which alternately access the processor, in order to avoid the conflicts which occur in the multitasking mode. Here, the semaphore technique is frequently used for the synchronization of the individual tasks. According to its mathematical-theoretical definition, a semaphore is an integral, non-negative variable associated with a queue. Here, the initial value of the semaphore defines how many tasks can be located simultaneously in a secured section controlled by a semaphore. The queue contains the tasks which wait for the secured section to be entered. To do this, a semaphore is checked and modified by the currently running task in order to implement the secure access to a variable via an uninterruptible read/write cycle. If, for example, this semaphore is greater than zero, it is decremented and the secure access to the desired variable is subsequently carried out by the currently running task. If the semaphore is already equal to zero, the task which requests a secure access is changed into the waiting state and the semaphore variable is not changed. At the end of the secure access to the variable, it is checked whether tasks are waiting on this semaphore, and if appropriate, one of the tasks located in the waiting state is activated; i.e., the processor is assigned. If there is no task waiting on the semaphore, the semaphore is incremented again by an uninterruptible read/write cycle. These uninterruptible read/write cycles to the semaphore variable can be implemented, in a way similar to the method of the task change inhibit, by a supervisor call and the subsequent handling by the operating system or in the user mode with special support by the processor hardware and processor bus hardware. Here too, time monitoring, whose function consists in avoiding the processor being blocked for a longer than average time, is provided for the duration of the secure access.

In the previously described implementations of a secure access to variables, a number of operating mode changes including the associated technical operating task processing or special support by processor hardware and processor bus hardware are

necessary during each access; i.e., secure accesses to variables increase the loading on the processor or require additional and specially supporting hardware.

An object to which the present invention is directed lies in improving the implementation of a secure access to at least one variable in a preemptively multitasking-controlled processor system.

SUMMARY OF THE INVENTION

An aspect of the method according to the present invention is that an access status memory is provided in a preemptively multitasking-controlled processor system for secure access to at least one variable, into which access status memory a blocking information item is input by the accessing task before a current access to at least one variable. Furthermore, when there is a task change intended by the task scheduler during the current access, the task scheduler checks the access status memory for a blocking information item which has been input and when the blocking information item has been input the task scheduler delays the intended task change. Finally, the task change information item is input into the access status memory using the blocking information item. At the end of the current access, a release information item is input into the access status memory by the currently accessing task and when a task change information item is input the requested task change is initiated by the currently accessing task. The use of an additional access status memory has the advantage that the switching over of the processor into the supervisor mode which, for example, is necessary with the task changing inhibit method, and the subsequent execution of an operating system task are dispensed with, and a considerable dynamic relieving of the loading on the processor is thus achieved, especially since secure accesses to variables occur very frequently when certain problems which occur during the operation of an information processing system arise. In addition, the inputting of the blocking information item, the task change information item or the release information item requires only a few machine instructions and is thus easy to implement in terms of programming technology. Furthermore, in the method according to the present invention, in contrast to the semaphore technique, no additional hardware support in the form of processor hardware or processor bus hardware is necessary, which leads to a cost-effective implementation of the secure access to variables which is not tied to specific hardware. Furthermore, during the secure access the accessing task is

advantageously not interrupted by a task change which is intended by a further task, and in addition the intended task change is not rejected but rather delayed so that after the evaluation of the task change information item at the end of the secure access the intended task change can be directly retrieved by the task scheduler.

A further aspect of the method according to the present invention is that, in addition to inputting the task change information item, a time monitoring system with a time period of at least the duration of the secure access is activated, and that the current access is terminated after the expiration of the defined time period. The time monitoring system in the method according to the present invention is not generally activated during the initialization of a secure access but rather only when there is a task change intended during the current access, and the dynamic loading, which is usually necessary during the use of the already known methods, for example semaphore technique or the setting of a task change inhibit, is thus dispensed with. This leads to an additional dynamic relieving of the load on the information processing system or the processor.

According to a further embodiment of the method according to the present invention, the contents of the access status memory are checked at the end of the secure access and before the inputting of the release information item so that when a task change information item is present, the activated time monitoring system is deactivated and a technical operating information item which initiates the intended task change is transmitted to the task scheduler by the currently accessing task. The checking of the contents of the access status memory advantageously ensures that, directly after termination of the secure access, the task scheduler is informed about the intended task change which is indicated by the task change information item, because without the indication of the technical operating information item which indicates the intended task change, the task scheduler would not carry out the delayed task change. Instead, the intended task change would be carried out at the time at which the currently accessing task is interrupted by the task scheduler; i.e., the intended task change would be unnecessarily delayed beyond the time period of the secure access.

Additional features and advantages of the present invention are described in, and will be apparent from, the following Detailed Description of the Preferred Embodiments and the Drawings.

DESCRIPTION OF THE DRAWINGS

Figure 1 shows a schematic diagram of an information processing system to which the method of the present invention is directed.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

In Figure 1, a first and a second user task T1, T2 and an operating system task BST are represented, by way of example, according to their processing over time by the processor of an information processing system which acts according to the preemptive multitasking method. Furthermore, a supervisor mode SM and a user mode UM of the processor and the associated tasks are indicated by two separate areas. Here, in the supervisor mode SM, the operating system task BST, later also called scheduler or task scheduler BST, is represented for processing by the processor, and in the user mode a first and a second user task T1, T2 for the processing by the processor are illustrated by way of example. A task which is currently in the waiting state, for example the operating system task BST at the time zero in Figure 1 and the second user task T2, is indicated using a broken line designated by BST and T2, and a currently executed task, the first user task T1 at the time zero in Figure 1, is indicated by an unbroken line designated by T1.

In order to represent the timing sequence of the method according to the invention of a secure access gz to at least one variable, a time axis t is provided on which a first, second, third, fourth and fifth time t1, t2, t2', t3, t3' are marked. Furthermore, a memory unit SE1 with an access status memory unit ZSE1 at the first, third and fourth time t1, t2', t3 is illustrated, information relating to the first, currently running task T1 being input in the memory unit ZSE1, and the memory can be implemented, for example, as part of a volatile memory. According to the method of the present invention, inter alia, a blocking information item SI, a task change information item WI and a release information item FI can be input into the access status memory unit ZSE1 which is assigned to the first, currently running user task T1.

Furthermore, the duration of a secure access gz to at least one variable by the first user task T1, which extends from the first time t1 to the fourth time t3, is illustrated. At the time zero, the first user task T1 is already currently assigned to the processor and the second user task T2 and the operating system task BST are in the waiting state. At the first time t1, the first user task T1 initializes a secure access to at

least one variable; i.e., the blocking information item SI is input into the access status memory unit ZSE1(t1) by the first user task T1 instead of the release information item FI which is input into it. Then, the first, currently executed user task T1 is in an uninterruptible execution state and can, thus, access the desired variables in a secured fashion.

At a later, second time t2, a task change request TWA is indicated as a result, for example, of an external event EE, for example the presence of external messages or as a result of the time period which is assigned to the first user task T1 by the task scheduler BST being exceeded, and the currently executed, first user task T1 is then changed into a quasi-waiting state wz by the task scheduler BST. Then, before the task scheduler BST initiates a task change TW after the task change request TWA has been received, the task scheduler BST checks the contents of the access status memory unit ZSE1(t2'). If a blocking information item SI relating to a third time t2' is input in the access memory unit ZSE1(t2') for the currently executed, first user task T1, the requested task change TWA is delayed by the task scheduler BST and instead of the blocking information SI a task change information item WI is input into the access status memory unit ZSE1(t2'). Then, the first, currently executed user task T1 is further processed and the quasi-waiting state wz is thus terminated again by the task scheduler BST. The first user task T1 can thus carry on the secure access (gz) for the desired variables without it being forced to release the processor by the task scheduler BST. In addition, at the third time t2', the task scheduler BST activates a time monitoring system TM in order to avoid the processor being blocked by the secure access gz of the first user task T1 for an unacceptably long time.

At the end of the secure access gz, indicated by way of example in Figure 1 as the fourth time t3, the contents of the access status memory unit ZSE1(t3) are firstly checked for the presence of a task change information item WI. If no task change information item WI has been input in the access status memory unit ZSE1(t3), the currently accessing, first user task T1 inputs the release information item FI instead of the present blocking information item SI, and the secure access gz is thus terminated; i.e., the currently accessing, first user task T1 can then be interrupted again. The currently accessing, first user task T1 can then access the processor until the task scheduler BST provides for a task change TW; i.e., the time of use of the processor

which is assigned to the first user task T1 by the task scheduler BST has expired or a task change request TWA is indicated to the task scheduler BST by an external event EE.

If, on the other hand, a task change information item WI is input, a task change request TWA is directly indicated to the task scheduler BST, as illustrated in Figure 1, so that, after the processing of the associated technical operating tasks, it can be used to carry out a task change TW. In addition, the release information item FI is input into the access status memory unit ZSE1(t3) by the first user task T1 instead of the input task change information item WI and after the secure access gz has been terminated, the time monitoring system TM is deactivated. Furthermore, the task scheduler BST which is executed in the supervisor mode SM extracts the processor from the first user task T1 and changes it to the waiting state.

Then, in the time period between the fourth and the fifth times t3, t3', the technical operating tasks which are provided by the task scheduler BST for a task change TW are processed within the supervisor mode; i.e., a task change TW is carried out by the operating system. For the execution of the second user task T2 which the processor has assigned at that particular time, the processor is switched over into the user mode and the second user task T2 can thus be assigned to the processor starting from the fifth time t3'.

Although the present invention has been described with reference to specific embodiments, those of skill in the art will recognize that changes may be made thereto without departing from the spirit and scope of the invention as set forth in the hereafter appended claims.

ABSTRACT OF THE DISCLOSURE

A method for secure access to at least one variable in a preemptively multitasking-controlled processor system wherein a blocking information item is input into an access status memory by an accessing task before a current access to at least one variable, and when there is a task change intended by a task scheduler during the secured current access a task change information item is input into the access status memory using the task scheduler. At the end of the current access, a release information item is input into the access status memory and the delayed task change is

initiated by the currently accessing task when a task change information item has been input.

In the claims:

On page 12, cancel line 1, and substitute the following left-hand justified heading therefor:

I Claim as My Invention:

Please cancel claims 1-6, without prejudice, and substitute the following claims therefor:

7. A method for secure access to at least one variable in a preemptively multitasking-controlled processor system, the method comprising the steps of:

providing a task scheduler for processing tasks;

providing an access status memory;

inputting, via an accessing task, a blocking information item into the access status memory before the secure access to the at least one variable;

checking, via the task scheduler and when there is a task change intended by the task scheduler during the secure access, the access status memory for an input blocking information item;

delaying the intended task change via the task scheduler when the blocking information item is input;

inputting a task change information item using the input blocking information item;

inputting, via the currently accessing task, a release information item into the access status memory at the end of the secure access; and

initiating the intended task change, via the currently accessing task, when the task change information item is input.

8. A method for secure access to at least one variable in a preemptively multitasking-controlled processor system as claimed in claim 7, the method further comprising the steps of:

activating a time monitoring system having a time period of at least a duration of the secure access; and

terminating the secure access after the expiration of the defined time period.

9. A method for secure access to at least one variable in a preemptively multitasking-controlled processor system as claimed in claim 8, the method further comprising the steps of:

checking contents of the access status memory at the end of the secure access and before the inputting of the release information item; and

deactivating the activated time monitoring system when the task change information item is present and transmitting a technical operating information item which initiates the intended task change to the task scheduler by the currently accessing task.

10. A method for secure access to at least one variable in a preemptively multitasking-controlled processor system as claimed in claim 7, the method further comprising the step of:

overwriting contents of the access status memory by the inputting of at least one of the blocking information item, the task change information item and the release information item into the access status memory.

11. A method for secure access to at least one variable in a preemptively multitasking-controlled processor system as claimed in claim 7, the method further comprising the step of:

forming the blocking information item, the task change information item and the release information item by at least one single-bit information item.

12. A method for secure access to at least one variable in a preemptively multitasking-controlled processor system as claimed in claim 7, the method further comprising the step of:

representing a variable by one of a variable of a software module which is stored in a memory unit and a hardware-related setting information item which is stored in a hardware register.

REMARKS

The present amendment makes editorial changes and corrects typographical errors in the specification, which includes the Abstract, in order to conform the specification to the requirements of United States Patent Practice. No new matter is added thereby. Attached hereto is a marked-up version of the changes made to the specification by the present amendment. The attached page is captioned "**Version With Markings To Show Changes Made".**

In addition, the present amendment cancels original claims 1-6 in favor of new claims 7-12. Claims 7-12 have been presented solely because the revisions by crossing out underlining which would have been necessary in claims 1-6 in order to present those claims in accordance with preferred United States Patent Practice would have been too extensive, and thus would have been too burdensome. The present amendment is intended for clarification purposes only and not for substantial reasons related to patentability pursuant to 35 U.S.C. §§103, 102, 103 or 112. Indeed, the cancellation of claims 1-6 does not constitute an intent on the part of the Applicants to surrender any of the subject matter of claims 1-6.

Early consideration on the merits is respectfully requested.

Respectfully submitted,

________________________ (Reg. No. 39,056)

William E. Vaughan
Bell, Boyd & Lloyd LLC
P.O. Box 1135
Chicago, Illinois 60690-1135
(312) 807-4292
Attorneys for Applicants

VERSIONS WITH MARKINGS TO SHOW CHANGES MADE

In The Specification:

The Specification of the present application, including the Abstract, has been amended as follows:

SPECIFICATION

TITLE

~~Method for secure access to at least one variable~~

~~in a preemptively multitasking-controlled processor system~~

MULTITASKING-CONTROLLED PROCESSOR SYSTEM

BACKGROUND OF THE INVENTION

~~Description~~

Field of the Invention

In existing and future information processing systems, ~~for example~~ such as personal computers, software objects ~~–usually~~ (also referred to as processes) - are and will be administered using the operating system in such a way that the hardware system, in particular the process-processing device which is provided in the information processing system, ~~for example~~ such as the processor, is utilized uniformly with the aim of high overall efficiency. In this way, the software modules which are assigned to the processor by the operating system – (~~usually~~ also referred to as tasks) - are processed by the processor. Here, special operating systems, for example Windows 95, are provided for the information processing systems which have a monoprocessor, i.e. the information processing system has just one processor, ~~said~~ the operating systems also permitting multi-user operation or multiple-process operation on a monoprocessor - see in this respect ~~in particular~~ "Architektur von Betriebssystemen" [Architecture of Operating Systems], H. Wetterstein, Hanser Studien Bücher [publishing house], 1984, pp. 54 et seq. The operating mode which is required for the multiple-process operation of a processor is known in the specialist field under the term "multiprogramming" or else "multitasking". In this way, during the execution of a task the information processing system can also carry out a further task such as the reading of data from a storage medium of the information processing system or, for example, the displaying of data on a data viewing station in a "quasiparallel" fashion.

Furthermore, a distinction is made between "cooperative" and "preemptive" multitasking. In the case of "cooperative" multitasking, each individual currently executed task itself determines, according to requirements, the time period for which it takes up the processor~~,~~; i.e., the currently running task decides on the time when the processor is released for the processing of further tasks. In the case of "preemptive" multitasking, a task of the operating system, known in the specialist field as "scheduler"~~,~~ or even "task scheduler", interrupts the currently executed task after a predefined or assigned time period has finished~~,~~; i.e., the time when the processor is assigned and released is determined using the task scheduler.

In order to execute a function of the operating system, ~~i.e.~~ for example an operating system task such as the task scheduler, a special operating mode of the processor for protecting the data of the operating system task is provided which is known as supervisor or kernel mode - see ~~in particular~~ Andrew S. Tanenbaum, "Betriebssysteme - Entwurf und Realisierung" [Operating Systems - Design and Implementation] part 1, Prentice- Hall International, 1990, pp 31/32. To do this, the processor is switched over using a supervisor call from a user mode into the supervisor mode and the control of the processor is thus transferred to the operating system or its tasks. In contrast with the supervisor mode, not all instructions are acceptable in the user mode, inter alia, in the user mode the use of input and output instructions and of some special instructions is prohibited. Likewise, in the user mode the access to all the data is generally not possible, ~~i.e.~~ for example the data of the operating system can neither be read nor amended for non-operating system tasks.

Specifically in the case of information processing systems which act according to the multitasking principle, variables or blocks of variables which are accessed during the processing of a task must be protected against competing accesses, for example by further tasks. This ensures that, for example, the errors occurring during dual simultaneous variable access cannot lead to any blockages of further tasks or of the entire information processing system. Such a protection mechanism is described below using the formulation "secured access" to at least one variable, and the term variable can refer here both to a variable of a software module which is stored in a memory unit and to a hardware-related setting information item which is stored in a hardware register. Such secured accesses frequently take place when specific problems

are posed, for example in information systems which are used to control real time systems but must also access data which can be administrated, and are of short duration in comparison to the average time period between two successive task changes. Consequently, the probability of a task change during a secure access is very low, but cannot at all be excluded.

The implementation of a "secure access" by a task can be carried out using various protection mechanisms. This includes, inter alia, the setting of a task change inhibit in order to avoid a competing access by a further task to the variables which are being accessed by the task currently running on the processor. To do this, before the variables to be read are accessed using a supervisor call, the processor is switched over into the supervisor mode and the setting of a task change inhibit is requested from the operating system in order to obtain exclusive access for the processor, and thus also for the desired variable, for the currently accessing task. Then, the processor is switched back into the user mode and the desired access to the variable can be secured by the previously interrupted task~~,~~; i.e., without interruption. After termination of the secure access by the currently running task, it is necessary to change again into the supervisor mode ~~by means of~~ via a supervisor call and for the task change inhibit to be reset by the operating system in ~~said~~ the supervisor mode. In order to further process the task which is currently to be processed, the processor is then changed back into the user mode and the time monitoring activated during the setting of the task change inhibit is deactivated in order to avoid the processor being blocked for an indeterminately long time.

A further method of implementing a secure access is used in the synchronization of tasks, i.e. the coordination of a ~~plurality~~ number of tasks which alternately access the processor, in order to avoid the conflicts which occur in the multitasking mode. Here, the semaphore technique is frequently used for the synchronization of the individual tasks. According to its mathematical-theoretical definition, a semaphore is an integral, non-negative variable associated with a queue. Here, the initial value of the semaphore defines how many tasks can be located simultaneously in a secured section controlled by a semaphore. The queue contains the tasks which wait for the secured section to be entered. To do this, a semaphore is checked and modified by the currently running task in order to implement the secure

access to a variable ~~by means of~~ via an uninterruptible read/write cycle. If, for example, this semaphore is greater than zero, it is decremented and the secure access to the desired variable is subsequently carried out by the currently running task. If the semaphore is already equal to zero, the task which requests a secure access is changed into the waiting state and the semaphore variable is not changed. At the end of the secure access to the variable, it is checked whether tasks are waiting on this semaphore, and if appropriate, one of the tasks located in the waiting state is activated~~,~~; i.e., the processor is assigned. If there is no task waiting on the semaphore, the semaphore is incremented again by ~~means of~~ an uninterruptible read/write cycle. These uninterruptible read/write cycles to the semaphore variable can ~~either~~ be implemented, in a way similar to the method of the task change inhibit, by a supervisor call and the subsequent handling by the operating system or in the user mode with special support by the processor hardware and processor bus hardware. Here too, time monitoring, whose function consists in avoiding the processor being blocked for a longer than average time, is provided for the duration of the secure access.

In the previously described implementations of a secure access to variables, a ~~plurality~~ number of operating mode changes including the associated technical operating task processing or special support by processor hardware and processor bus hardware are necessary during each access~~,~~; i.e., secure accesses to variables increase the loading on the processor or require additional and specially supporting hardware.

~~The~~ An object ~~on~~ to which the present invention is ~~based consists~~ directed lies in improving the implementation of a secure access to at least one variable in a preemptively multitasking-controlled processor system. ~~The object is achieved by means of the features of patent claim 1.~~

SUMMARY OF THE INVENTION

~~The essential~~ An aspect of the method according to the present invention is that an access status memory is provided in a preemptively multitasking-controlled processor system for secure access to at least one variable, into which access status memory a blocking information item is input by the accessing task before a current access to at least one variable. Furthermore, when there is a task change intended by the task scheduler during the current access, the task scheduler checks the access status memory for a blocking information item which has been input and when the blocking

information item has been input the task scheduler delays the intended task change. Finally, the task change information item is input into the access status memory using ~~said~~ the blocking information item. At the end of the current access, a release information item is input into the access status memory by the currently accessing task ~~(T1)~~ and when a task change information item is input the requested task change is initiated by the currently accessing task ~~(t1)~~. The use of an additional access status memory has the advantage that the switching over of the processor into the supervisor mode~~,~~ which, for example, is necessary with the task changing inhibit method, and the subsequent execution of an operating system task are dispensed with, and a considerable dynamic relieving of the loading on the processor is thus achieved, especially since secure accesses to variables occur very frequently when certain problems which occur during the operation of an information processing system arise. In addition, the inputting of the blocking information item, the task change information item or the release information item requires only a few machine instructions and is thus easy to implement in terms of programming technology. Furthermore, in the method according to the present invention, in contrast to the semaphore technique, no additional hardware support in the form of processor hardware or processor bus hardware is necessary, which leads to a cost-effective implementation of the secure access to variables which is not tied to specific hardware. Furthermore, during the secure access the accessing task is advantageously not interrupted by a task change which is intended by a further task, and in addition the intended task change is not rejected but rather delayed so that after the evaluation of the task change information item at the end of the secure access the intended task change can be directly retrieved by the task scheduler.

A further ~~essential~~ aspect of the method according to the present invention is that, in addition to inputting the task change information item, a time monitoring system with a time period ~~comprising~~ of at least the duration of the secure access is activated, and that the current access is terminated after the ~~expiry~~ expiration of the defined time period ~~-claim 2~~. The time monitoring system in the method according to the present invention is ~~advantageously~~ not generally activated during the initialization of a secure access but rather only when there is a task change intended during the current access, and the dynamic loading, which is usually necessary during the use of

the already known methods, for example semaphore technique or the setting of a task change inhibit, is thus dispensed with. This leads to an additional dynamic relieving of the load on the information processing system or the processor.

According to a further ~~refinement~~ embodiment of the method according to the present invention, the contents of the access status memory are checked at the end of the secure access and before the inputting of the release information item so that when a task change information item is present, the activated time monitoring system is deactivated and a technical operating information item which initiates the intended task change is transmitted to the task scheduler by the currently accessing task ~~-claim 3~~. The checking of the contents of the access status memory advantageously ensures that, directly after termination of the secure access, the task scheduler is informed about the intended task change which is indicated by the task change information item, because without the indication of the technical operating information item which indicates the intended task change, the task scheduler would not carry out the delayed task change. Instead, the intended task change would be carried out at the time at which the currently accessing task is interrupted by the task scheduler~~,~~; i.e., the intended task change would be unnecessarily delayed beyond the time period of the secure access.

~~Further advantageous refinements of the method according to the invention can be found in the further claims.~~

~~The method according to the invention will be explained in more detail below with reference to a figure.~~

Additional features and advantages of the present invention are described in, and will be apparent from, the following Detailed Description of the Preferred Embodiments and the Drawings.

DESCRIPTION OF THE DRAWINGS

Figure 1 shows a schematic diagram of an information processing system to which the method of the present invention is directed.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

In Figure 1, a first and a second user task T1, T2 and an operating system task BST are represented, by way of example, according to their processing over time by the processor of an information processing system which acts according to the preemptive multitasking method. Furthermore, a supervisor mode SM and a user

mode UM of the processor and the associated tasks are indicated by two separate areas. Here, in the supervisor mode SM, the operating system task BST, later also called scheduler or task scheduler BST, is represented for processing by the processor, and in the user mode a first and a second user task T1, T2 for the processing by the processor are illustrated by way of example. A task which is currently in the waiting state, - for example ~~in particular~~ the operating system task BST at the time zero in ~~f~~Figure 1~~,~~ and the second user task T2, - is indicated using a broken line designated by BST and T2, and a currently executed task, - the first user task T1 at the time zero in Figure 1, - is indicated by an unbroken line designated by T1.

In order to represent the timing sequence of the method according to the present invention of a secure access gz to at least one variable, a time axis t is provided on which a first, second, third, fourth and fifth time t1, t2, t2', t3, t3' are marked. Furthermore, a memory unit SE1 with an access status memory unit ZSE1 at the first, third and fourth time t1, t2', t3 is illustrated, information relating to the first, currently running task T1 being input in the memory unit ZSE1, and the memory can be implemented, for example, as part of a volatile memory. According to the method ~~according to~~ of the present invention, inter alia, a blocking information item SI, a task change information item WI and a release information item FI can be input into the access status memory unit ZSE1 which is assigned to the first, currently running user task T1.

Furthermore, the duration of a secure access gz to at least one variable by the first user task T1, which extends from the first time t1 to the fourth time t3, is illustrated. At the time zero, the first user task T1 is already currently assigned to the processor and the second user task T2 and the operating system task BST are in the waiting state. At the first time t1, the first user task T1 initializes a secure access to at least one variable~~,~~; i.e., the blocking information item SI is input into the access status memory unit ZSE1(t1) by the first user task T1 instead of the release information item FI which is input into it. Then, the first, currently executed user task T1 is in an uninterruptible execution state and can, thus, access the desired variables in a secured fashion.

At a later, second time t2, a task change request TWA is indicated as a result, for example, of an external event EE, for example the presence of external messages or

as a result of the time period which is assigned to the first user task T1 by the task scheduler BST being exceeded, and the currently executed, first user task T1 is then changed into a quasi-waiting state wz by the task scheduler BST. Then, before the task scheduler BST initiates a task change TW after the task change request TWA has been received, ~~said~~ the task scheduler BST checks the contents of the access status memory unit ZSE1(t2'). If a blocking information item SI relating to a third time t2' is input in the access memory unit ZSE1(t2') for the currently executed, first user task T1, the requested task change TWA is delayed by the task scheduler BST and instead of the blocking information SI a task change information item WI is input into the access status memory unit ZSE1(t2'). Then, the first, currently executed user task T1 is further processed and the quasi-waiting state wz is thus terminated again by the task scheduler BST. The first user task T1 can thus carry on the secure access (gz) for the desired variables without it being forced to release the processor by the task scheduler BST. In addition, at the third time t2', the task scheduler BST activates a time monitoring system TM in order to avoid the processor being blocked by the secure access gz of the first user task T1 for an unacceptably long time.

At the end of the secure access gz, - indicated by way of example in Figure 1 as the fourth time t3, - the contents of the access status memory unit ZSE1(t3) are firstly checked for the presence of a task change information item WI. If no task change information item WI has been input in the access status memory unit ZSE1(t3), the currently accessing, first user task T1 inputs the release information item FI instead of the present blocking information item SI, and the secure access gz is thus terminated~~,~~; i.e., the currently accessing, first user task T1 can then be interrupted again. The currently accessing, first user task T1 can then access the processor until the task scheduler BST provides for a task change TW~~,~~; i.e., the time of use of the processor which is assigned to the first user task T1 by the task scheduler BST has expired or a task change request TWA is indicated to the task scheduler BST by an external event EE.

If, on the other hand, a task change information item WI is input, a task change request TWA is directly indicated to the task scheduler BST, - as illustrated in Figure 1, - so that, after the processing of the associated technical operating tasks, it can be used to carry out a task change TW. In addition, the release information item FI

is input into the access status memory unit ZSE1(t3) by the first user task T1 instead of the input task change information item WI and after the secure access gz has been terminated, the time monitoring system TM is deactivated. Furthermore, the task scheduler BST which is executed in the supervisor mode SM extracts the processor from the first user task T1 and changes it to the waiting state.

Then, in the time period between the fourth and the fifth times t3, t3', the technical operating tasks which are provided by the task scheduler BST for a task change TW are processed within the supervisor mode~~,~~; i.e., a task change TW is carried out by the operating system. For the execution of the second user task T2 which the processor has assigned at that particular time, the processor is switched over into the user mode and the second user task T2 can thus be assigned to the processor starting from the fifth time t3'.

Although the present invention has been described with reference to specific embodiments, those of skill in the art will recognize that changes may be made thereto without departing from the spirit and scope of the invention as set forth in the hereafter appended claims.

~~Abstract~~

ABSTRACT OF THE DISCLOSURE

A ~~M~~method for secure access to at least one variable in a preemptively multitasking-controlled processor system wherein ~~A~~a blocking information item ~~(SI)~~ is input into an access status memory ~~(ZSE1)~~ by ~~the~~ an accessing task ~~(T1)~~ before a current access to at least one variable~~.~~, ~~Furthermore,~~ and when there is a task change intended by a task scheduler ~~(BST)~~ during the secured~~,~~ current access a task change information item ~~(WI)~~ is input into the access status memory ~~(ZSE1)~~ using the task scheduler ~~(BST)~~. At the end of the current access, a release information item ~~(FI)~~ is input into the access status memory ~~(ZSE1)~~ and the delayed task change ~~(TWA)~~ is initiated by the currently accessing task ~~(T1)~~ when a task change information item ~~(WI)~~ has been input.

~~Figure~~

BOX PCT

IN THE UNITED STATES ELECTED/DESIGNATED OFFICE
OF THE UNITED STATES PATENT AND TRADEMARK OFFICE
UNDER THE PATENT COOPERATION TREATY-CHAPTER II

SUBMISSION OF DRAWINGS

APPLICANT:	Dr. Gerhard Spitz	DOCKET NO: 112740-271
SERIAL NO:		GROUP ART UNIT:
		EXAMINER:
INTERNATIONAL APPLICATION NO:		PCT/DE00/00077
INTERNATIONAL FILING DATE:		11 January 2000
INVENTION:	A METHOD FOR SECURING ACCESS TO AT LEAST ONE VARIABLE IN A PREEMPTIVELY MULTITASKING-CONTROLLED PROCESSOR SYSTEM	

Assistant Commissioner for Patents,
Washington, D.C. 20231

Sir:

Applicant herewith submits one sheet (Fig. 1) of drawings for the above-referenced PCT application.

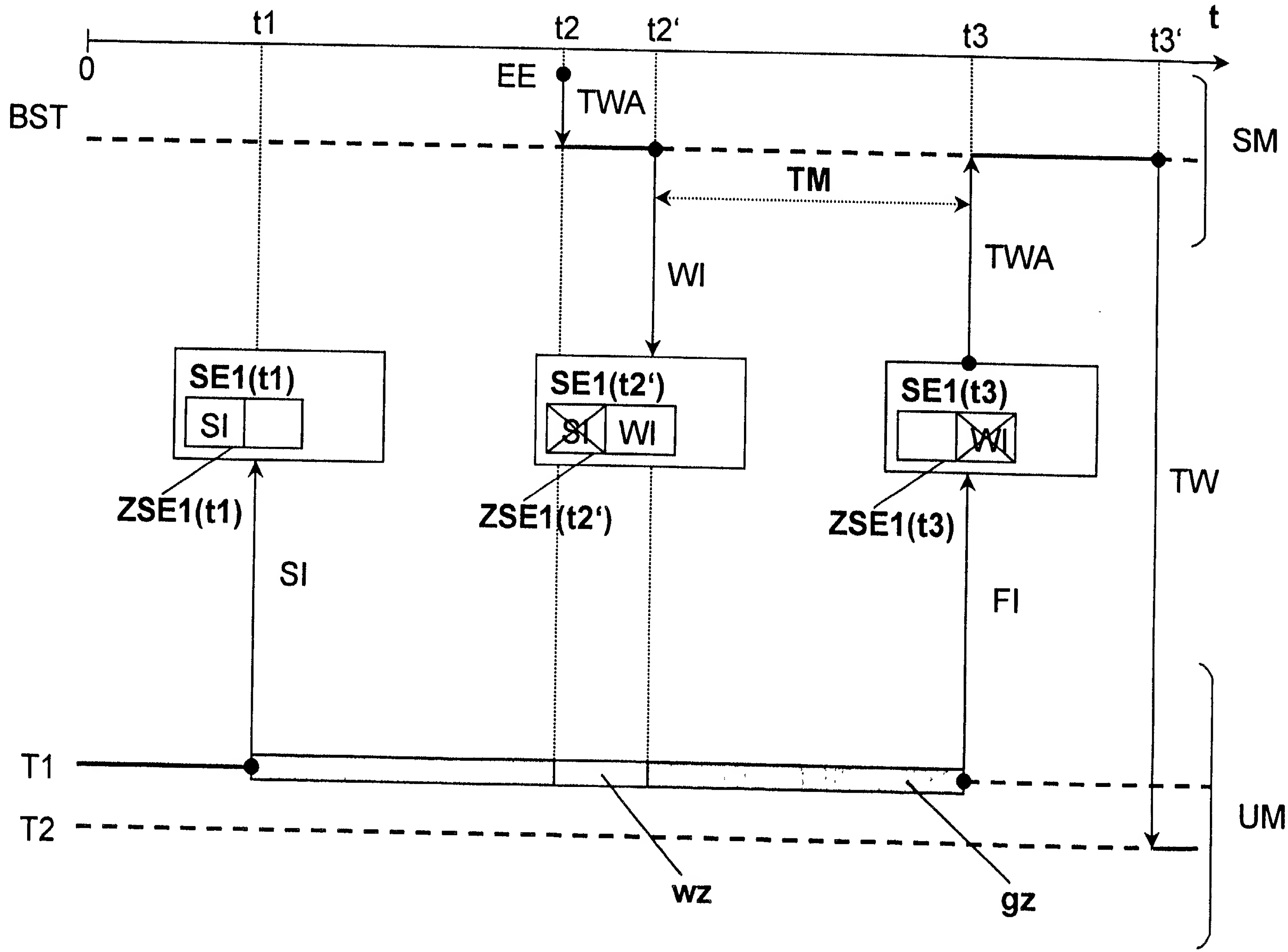
Respectfully submitted,

(Reg. No. 39,056)

William E. Vaughan
Bell, Boyd & Lloyd LLC
P.O. Box 1135
Chicago, Illinois 60690-1135
(312) 807-4292
Attorneys for Applicant

09/890432

FIG 1



JC17 Rec'd PCT/PTO 30 JUL 2001
09/890432

Description

Method for secure access to at least one variable in a preemptively multitasking-controlled processor system

In existing and future information processing systems, for example personal computers, software objects - usually also referred to as processes - are and will be administered using the operating system in such a way that the hardware system, in particular the process-processing device which is provided in the information processing system, for example the processor, is utilized uniformly with the aim of high overall efficiency. In this way, the software modules which are assigned to the processor by the operating system - usually also referred to as tasks - are processed by the processor. Here, special operating systems, for example Windows 95, are provided for the information processing systems which have a monoprocessor, i.e. the information processing system has just one processor, said operating systems also permitting multi-user operation or multiple-process operation on a monoprocessor - see in this respect in particular "Architektur von Betriebssystemen" [Architecture of Operating Systems], H. Wetterstein, Hanser Studien Bücher [publishing house], 1984, pp. 54 et seq. The operating mode which is required for the multiple-process operation of a processor is known in the specialist field under the term "multiprogramming" or else "multitasking". In this way, during the execution of a task the information processing system can also carry out a further task such as the reading of data from a storage medium of the information processing system or for example the displaying of data on a data viewing station in a "quasiparallel" fashion.

Furthermore, a distinction is made between

"cooperative" and "preemptive" multitasking. In the case of "cooperative" multitasking, each individual currently executed task itself determines, according to requirements, the time period for which it takes up the processor, i.e. the

currently running task decides on the time when the processor is released for the processing of further tasks. In the case of "preemptive" multitasking, a task of the operating system, known in the specialist field as "scheduler", or even "task scheduler" interrupts the currently executed task after a predefined or assigned time period has finished, i.e. the time when the processor is assigned and released is determined using the task scheduler.

In order to execute a function of the operating system, i.e. for example an operating system task such as the task scheduler, a special operating mode of the processor for protecting the data of the operating system task is provided which is known as supervisor or kernel mode - see in particular Andrew S. Tanenbaum, "Betriebssysteme - Entwurf und Realisierung" [Operating Systems - Design and Implementation] part 1, Prentice-Hall International, 1990, pp 31/32. To do this, the processor is switched over using a supervisor call from a user mode into the supervisor mode and the control of the processor is thus transferred to the operating system or its tasks. In contrast with the supervisor mode, not all instructions are acceptable in the user mode, inter alia, in the user mode the use of input and output instructions and of some special instructions is prohibited. Likewise, in the user mode the access to all the data is generally not possible, i.e. for example the data of the operating system can neither be read nor amended for non-operating system tasks.

Specifically in the case of information processing systems which act according to the multitasking principle, variables or blocks of variables which are accessed during the processing of a task must be protected against competing accesses, for example by further tasks. This ensures that, for example, the errors occurring during dual simultaneous variable access cannot lead to any blockages of further tasks or of the entire

information processing system. Such a protection mechanism is described below using the formulation "secured access" to at least one variable, and the term variable can refer here both to a variable of a software module which is stored in a memory unit and to a hardware-related setting information item which is stored in a hardware register. Such secured accesses frequently take place when specific problems are posed, for example in information systems which are used to control real time systems but must also access data which can be administrated, and are of short duration in comparison to the average time period between two successive task changes. Consequently, the probability of a task change during a secure access is very low, but cannot at all be excluded.

The implementation of a "secure access" by a task can be carried out using various protection mechanisms. This includes, inter alia, the setting of a task change inhibit in order to avoid a competing access by a further task to the variables which are being accessed by the task currently running on the processor. To do this, before the variables to be read are accessed using a supervisor call, the processor is switched over into the supervisor mode and the setting of a task change inhibit is requested from the operating system in order to obtain exclusive access for the processor, and thus also for the desired variable, for the currently accessing task. Then, the processor is switched back into the user mode and the desired access to the variable can be secured by the previously interrupted task, i.e. without interruption. After termination of the secure access by the currently running task, it is necessary to change again into the supervisor mode by means of a supervisor call and for the task change inhibit to be reset by the operating system in said mode. In order to further process the task which is currently to be processed,

the processor is then changed back into the user mode and the time monitoring activated during the setting of the task change inhibit is deactivated in order to avoid the processor being blocked for an indeterminately long time.

A further method of implementing a secure access is used in the synchronization of tasks, i.e. the coordination of a plurality of tasks which alternately access the processor, in order to avoid the conflicts which occur in the multitasking mode. Here, the semaphore technique is frequently used for the synchronization of the individual tasks. According to its mathematical-theoretical definition, a semaphore is an integral, non-negative variable associated with a queue. Here, the initial value of the semaphore defines how many tasks can be located simultaneously in a secured section controlled by a semaphore. The queue contains the tasks which wait for the secured section to be entered. To do this, a semaphore is checked and modified by the currently running task in order to implement the secure access to a variable by means of an uninterruptible read/write cycle. If, for example, this semaphore is greater than zero, it is decremented and the secure access to the desired variable is subsequently carried out by the currently running task. If the semaphore is already equal to zero, the task which requests a secure access is changed into the waiting state and the semaphore variable is not changed. At the end of the secure access to the variable, it is checked whether tasks are waiting on this semaphore, and if appropriate, one of the tasks located in the waiting state is activated, i.e. the processor is assigned. If there is no task waiting on the semaphore, the semaphore is incremented again by means of an uninterruptible read/write cycle. These uninterruptible read/write cycles to the semaphore variable can either be implemented, in a way similar to

the method of the task change inhibit, by a supervisor call and the subsequent handling by the operating system or in the user mode with special

support by the processor hardware and processor bus hardware. Here too, time monitoring, whose function consists in avoiding the processor being blocked for a longer than average time, is provided for the duration of the secure access.

In the previously described implementations of a secure access to variables, a plurality of operating mode changes including the associated technical operating task processing or special support by processor hardware and processor bus hardware are necessary during each access, i.e. secure accesses to variables increase the loading on the processor or require additional and specially supporting hardware.

The object on which the invention is based consists in improving the implementation of a secure access to at least one variable in a preemptively multitasking-controlled processor system. The object is achieved by means of the features of patent claim 1.

The essential aspect of the method according to the invention is that an access status memory is provided in a preemptively multitasking-controlled processor system for secure access to at least one variable, into which access status memory a blocking information item is input by the accessing task before a current access to at least one variable. Furthermore, when there is a task change intended by the task scheduler during the current access, the task scheduler checks the access status memory for a blocking information item which has been input and when the blocking information item has been input the task scheduler delays the intended task change. Finally, the task change information item is input into the access status memory using said blocking information item. At the end of the current access, a release information item is input into the access status memory by the currently accessing task (T1) and when a task change information item is input

the requested task change is initiated by the currently accessing task (t1). The use of an additional access status memory has the advantage that the switching over of the processor into the supervisor mode, which, for example, is necessary with the task changing inhibit method, and the subsequent execution of an operating system task are dispensed with, and a considerable dynamic relieving of the loading on the processor is thus achieved, especially since secure accesses to variables occur very frequently when certain problems which occur during the operation of an information processing system arise. In addition, the inputting of the blocking information item, the task change information item or the release information item requires only a few machine instructions and is thus easy to implement in terms of programming technology. Furthermore, in the method according to the invention, in contrast to the semaphore technique, no additional hardware support in the form of processor hardware or processor bus hardware is necessary, which leads to a cost-effective implementation of the secure access to variables which is not tied to specific hardware. Furthermore, during the secure access the accessing task is advantageously not interrupted by a task change which is intended by a further task, and in addition the intended task change is not rejected but rather delayed so that after the evaluation of the task change information item at the end of the secure access the intended task change can be directly retrieved by the task scheduler.

A further essential aspect of the method according to the invention is that in addition to inputting the task change information item a time monitoring system with a time period comprising at least the duration of the secure access is activated, and that the current access is terminated after the expiry of the defined time period - claim 2. The time

monitoring system in the method according to the invention is advantageously not generally activated during the initialization of a secure access but rather only when there is a task change intended during the current access, and

the dynamic loading, which is usually necessary during the use of the already known methods, for example semaphore technique or the setting of a task change inhibit, is thus dispensed with. This leads to an additional dynamic relieving of the load on the information processing system or the processor.

According to a further refinement of the method according to the invention, the contents of the access status memory are checked at the end of the secure access and before the inputting of the release information item so that when a task change information item is present the activated time monitoring system is deactivated and a technical operating information item which initiates the intended task change is transmitted to the task scheduler by the currently accessing task - claim 3. The checking of the contents of the access status memory advantageously ensures that, directly after termination of the secure access, the task scheduler is informed about the intended task change which is indicated by the task change information item, because without the indication of the technical operating information item which indicates the intended task change the task scheduler would not carry out the delayed task change. Instead, the intended task change would be carried out at the time at which the currently accessing task is interrupted by the task scheduler, i.e. the intended task change would be unnecessarily delayed beyond the time period of the secure access.

Further advantageous refinements of the method according to the invention can be found in the further claims.

The method according to the invention will be explained in more detail below with reference to a figure.

The method according to the invention will be explained in more detail below with reference to a figure.

In Figure 1, a first and a second user task T1, T2 and an operating system task BST are represented by way of example according to their processing over time by the processor of an information processing system which acts according to the preemptive multitasking method. Furthermore, a supervisor mode SM and a user mode UM of the processor and the associated tasks are indicated by two separate areas. Here, in the supervisor mode SM the operating system task BST, later also called scheduler or task scheduler BST, is represented for processing by the processor, and in the user mode a first and a second user task T1, T2 for the processing by the processor are illustrated by way of example. A task which is currently in the waiting state - for example in particular the operating system task BST at the time zero in figure 1, and the second user task T2 - is indicated using a broken line designated by BST and T2, and a currently executed task - the first user task T1 at the time zero in Figure 1 - is indicated by an unbroken line designated by T1.

In order to represent the timing sequence of the method according to the invention of a secure access gz to at least one variable, a time axis t is provided on which a first, second, third, fourth and fifth time t1, t2, t2′, t3, t3′ are marked. Furthermore, a memory unit SE1 with an access status memory unit ZSE1 at the first, third and fourth time t1, t2′, t3 is illustrated, information relating to the first, currently running task T1 being input in the memory unit ZSE1, and the memory can be implemented, for example, as part of a volatile memory. According to the method according to the invention, inter alia, a blocking information item SI, a task change information item WI and a release information item FI can be input into the access status memory unit ZSE1 which is assigned to the first, currently running user task T1.

Furthermore, the duration of a secure access gz to at least one variable by the first user task T1, which extends from the first time t1 to the fourth time t3, is illustrated. At the time zero, the first user task T1 is already currently assigned to the processor and the second user task T2 and the operating system task BST are in the waiting state. At the first time t1, the first user task T1 initializes a secure access to at least one variable, i.e. the blocking information item SI is input into the access status memory unit ZSE1(t1) by the first user task T1 instead of the release information item FI which is input into it. Then, the first, currently executed user task T1 is in an uninterruptible execution state and can thus access the desired variables in a secured fashion.

At a later, second time t2, a task change request TWA is indicated as a result, for example, of an external event EE, for example the presence of external messages or as a result of the time period which is assigned to the first user task T1 by the task scheduler BST being exceeded and the currently executed, first user task T1 is then changed into a quasi-waiting state wz by the task scheduler BST. Then, before the task scheduler BST initiates a task change TW after the task change request TWA has been received, said task scheduler BST checks the contents of the access status memory unit ZSE1(t2'). If a blocking information item SI relating to a third time t2' is input in the access memory unit ZSE1(t2') for the currently executed, first user task T1, the requested task change TWA is delayed by the task scheduler BST and instead of the blocking information SI a task change information item WI is input into the access status memory unit ZSE1(t2'). Then, the first, currently executed user task T1 is further processed and the quasi-waiting state wz is thus terminated again by the task scheduler BST. The first user task T1 can thus

carry on the secure access (gz) for the desired variables without it being forced to release the processor by the task scheduler BST. In addition, at the third time t2′, the task scheduler BST activates a time monitoring system TM in order to avoid the processor being blocked by the secure access gz of the first user task T1 for an unacceptably long time.

At the end of the secure access gz - indicated by way of example in Figure 1 as the fourth time t3 - the contents of the access status memory unit ZSE1(t3) are firstly checked for the presence of a task change information item WI. If no task change information item WI has been input in the access status memory unit ZSE1(t3), the currently accessing, first user task T1 inputs the release information item FI instead of the present blocking information item SI, and the secure access gz is thus terminated, i.e. the currently accessing, first user task T1 can then be interrupted again. The currently accessing, first user task T1 can then access the processor until the task scheduler BST provides for a task change TW, i.e. the time of use of the processor which is assigned to the first user task T1 by the task scheduler BST has expired or a task change request TWA is indicated to the task scheduler BST by an external event EE.

If, on the other hand, a task change information item WI is input, a task change request TWA is directly indicated to the task scheduler BST - as illustrated in Figure 1 - so that, after the processing of the associated technical operating tasks, it can be used to carry out a task change TW. In addition, the release information item FI is input into the access status memory unit ZSE1(t3) by the first user task T1 instead of the input task change information item WI and after the secure access gz has been terminated the time monitoring system TM is deactivated. Furthermore, the task scheduler BST which is executed in the supervisor mode SM

extracts the processor from the first user task T1 and changes it to the waiting state.

Then, in the time period between the fourth and the fifth times t3, t3', the technical operating tasks which are provided by the task scheduler BST for a task change TW are processed within the supervisor mode, i.e. a task change TW is carried out by the operating system. For the execution of the second user task T2 which the processor has assigned at that particular time, the processor is switched over into the user mode and the second user task T2 can thus be assigned to the processor starting from the fifth time t3'.

Patent Claims

1. A method for secure access (gz) to at least one variable in a preemptively multitasking-controlled processor system, a task scheduler (BST) being provided for processing the tasks (T1, T2),
in which an access status memory (ZSE1) is provided

- into which a blocking information item (SI) is input by the accessing task (T1) before a current access (gz) to at least one variable,
- in which when there is a task change (TW) intended by the task scheduler (BST) during the current access (gz), the task scheduler (BST) checks the access status memory (ZSE1) for an input blocking information item (SI) and when the blocking information item (SI) is input the task scheduler (BST) delays the intended task change (TWA) and a task change information item (WI) is input using said blocking information item (SI), and
- into which a release information item (FI) is input by the currently accessing task (T1) at the end of the current access (gz), and when a task change information item (WI) is input the intended task change (TWA) is initiated by the currently accessing task (TI).

2. The method as claimed in claim 1, characterized in that in addition to inputting the task change information item (WI) a time monitoring system (TM) with a time period comprising at least the duration of the secure access (gz) is activated, and that the current access (gz) is terminated after the expiry of the defined time period.

3. The method as claimed in claim 2, characterized in that at the end of the secure access (gz) and before the inputting of the release information item (FI) the contents of the access status memory (ZSE) are checked so that when a

task change information item (WI) is present the activated time monitoring system (TM) is deactivated and a technical operating information item which initiates the intended task change is transmitted to the task scheduler (BST) by the currently accessing task (T1).

4. The method as claimed in one of claims 1 to 3, characterized in that the contents of the access status memory (ZSE1) are overwritten by the inputting of an information item (SI, WI, FI) into the access status memory (ZSE1).

5. The method as claimed in one of claims 1 to 4, characterized in that the blocking information item (SI), the task change information item (WI) and the enable information item (FI) are formed by at least one single-bit information item.

6. The method as claimed in one of claims 1 to 5, characterized in that a variable is represented either by a variable of a software module which is stored in a memory unit or by a hardware-related setting information item which is stored in a hardware register.

GR 99 P 1129

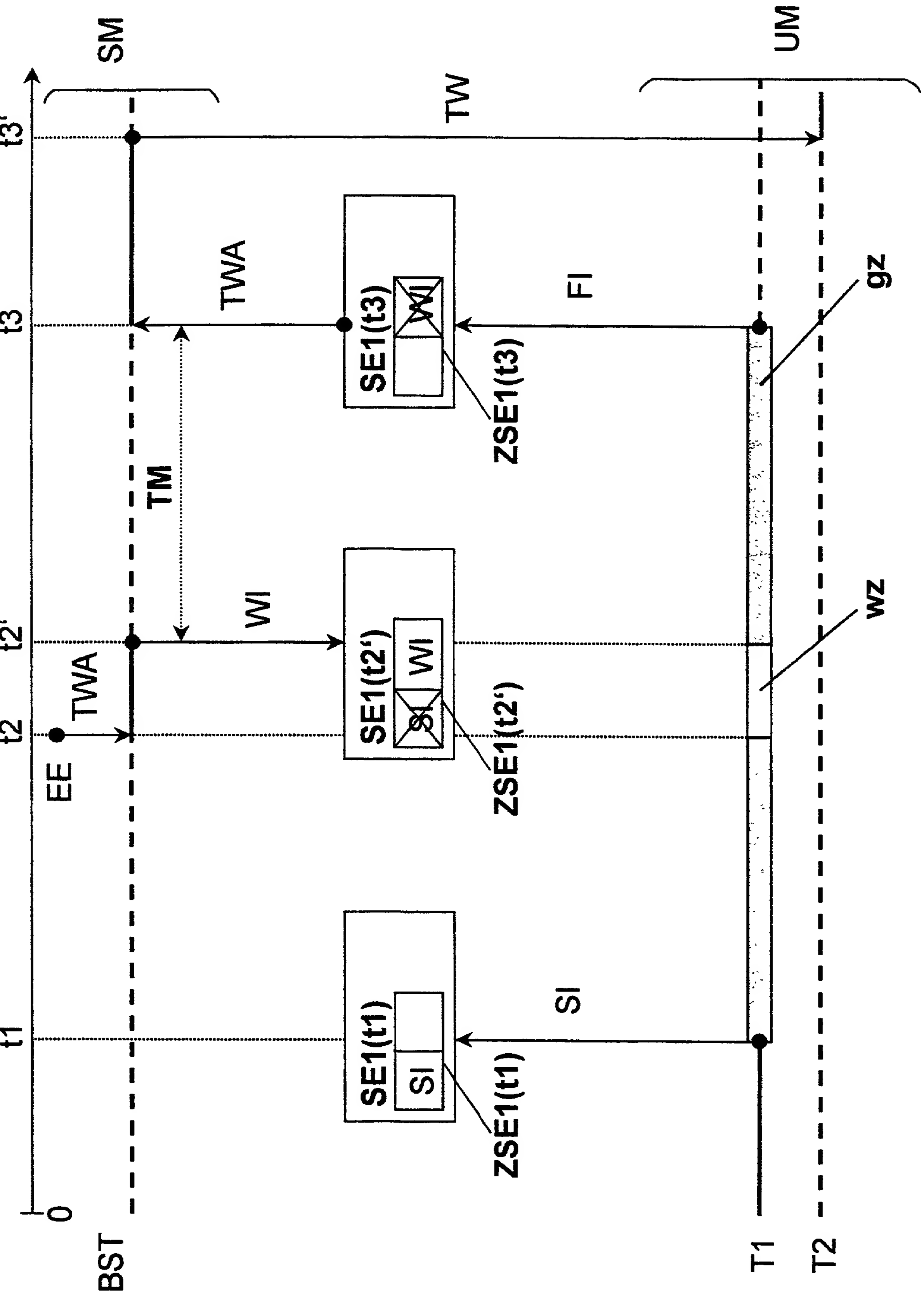
Abstract

Method for secure access to at least one variable in a preemptively multitasking-controlled processor system

A blocking information item (SI) is input into an access status memory (ZSE1) by the accessing task (T1) before a current access to at least one variable. Furthermore, when there is a task change intended by a task scheduler (BST) during the secured, current access a task change information item (WI) is input into the access status memory (ZSE1) using the task scheduler (BST). At the end of the current access, a release information item (FI) is input into the access status memory (ZSE1) and the delayed task change (TWA) is initiated by the currently accessing task (T1) when a task change information item (WI) has been input.

Figure

FIG 1


t
SM
UM
t3'
t3
t2'
t2
t1
0
BST
T1
T2
TW
TWA
TM
WI
EE
FI
SI
SE1(t3)
SE1(t2')
SE1(t1)
ZSE1(t3)
ZSE1(t2')
ZSE1(t1)
gz
wz

Declaration and Power of Attorney For Patent Application

Erklärung Für Patentanmeldungen Mit Vollmacht

German Language Declaration

Als nachstehend benannter Erfinder erkläre ich hiermit an Eides Statt:

As a below named inventor, I hereby declare that:

dass mein Wohnsitz, meine Postanschrift, und meine Staatsangehörigkeit den im Nachstehenden nach meinem Namen aufgeführten Angaben entsprechen,

My residence, post office address and citizenship are as stated below next to my name,

dass ich, nach bestem Wissen der ursprüngliche, erste und alleinige Erfinder (falls nachstehend nur ein Name angegeben ist) oder ein ursprünglicher, erster und Miterfinder (falls nachstehend mehrere Namen aufgeführt sind) des Gegenstandes bin, für den dieser Antrag gestellt wird und für den ein Patent beantragt wird für die Erfindung mit dem Titel:

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

VERFAHREN ZUM GESICHERTEN ZUGRIFF AUF ZUMINDEST EINE VARIABLE IN EINEM PRAEEMPTIV MULTITASKING-GESTEUERTEN PROZESSORSYSTEM

METHOD FOR PROTECTED ACCESS TO AT LEAST ONE VARIABLE IN A PREEMPTIVE MULTITASKING-CONTROLLED PROCESSOR SYSTEM

deren Beschreibung

(zutreffendes ankreuzen)
☐ hier beigefügt ist.
☒ am 11.01.2000 als
PCT internationale Anmeldung
PCT Anmeldungsnummer PCT/DE00/00077
eingereicht wurde und am ________
abgeändert wurde (falls tatsächlich abgeändert).

the specification of which

(check one)
☐ is attached hereto.
☒ was filed on 11.01.2000 as
PCT international application
PCT Application No. PCT/DE00/00077
and was amended on ________
(if applicable)

Ich bestätige hiermit, dass ich den Inhalt der obigen Patentanmeldung einschliesslich der Ansprüche durchgesehen und verstanden habe, die eventuell durch einen Zusatzantrag wie oben erwähnt abgeändert wurde.

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims as amended by any amendment referred to above.

Ich erkenne meine Pflicht zur Offenbarung irgendwelcher Informationen, die für die Prüfung der vorliegenden Anmeldung in Einklang mit Absatz 37, Bundesgesetzbuch, Paragraph 1.56(a) von Wichtigkeit sind, an.

I acknowledge the duty to disclose information which is material to the examination of this application in accordance with Title 37, Code of Federal Regulations, §1.56(a).

Ich beanspruche hiermit ausländische Prioritätsvorteile gemäss Abschnitt 35 der Zivilprozessordnung der Vereinigten Staaten, Paragraph 119 aller unten angegebenen Auslandsanmeldungen für ein Patent oder eine Erfindersurkunde, und habe auch alle Auslandsanmeldungen für ein Patent oder eine Erfindersurkunde nachstehend gekennzeichnet, die ein Anmeldedatum haben, das vor dem Anmeldedatum der Anmeldung liegt, für die Priorität beansprucht wird.

I hereby claim foreign priority benefits under Title 35, United States Code, §119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

IDNR: 2590 / V: 99-1.00 / B:Val

German Language Declaration

Prior foreign appplications
Priorität beansprucht

Priority Claimed

19903599.7	DE	29.01.1999	☒ Yes / Ja	☐ No / Nein
(Number) (Nummer)	(Country) (Land)	(Day Month Year Filed) (Tag Monat Jahr eingereicht)		
(Number) (Nummer)	(Country) (Land)	(Day Month Year Filed) (Tag Monat Jahr eingereicht)	☐ Yes / Ja	☐ No / Nein
(Number) (Nummer)	(Country) (Land)	(Day Month Year Filed) (Tag Monat Jahr eingereicht)	☐ Yes / Ja	☐ No / Nein

Ich beanspruche hiermit gemäss Absatz 35 der Zivilprozessordnung der Vereinigten Staaten, Paragraph 120, den Vorzug aller unten aufgeführten Anmeldungen und falls der Gegenstand aus jedem Anspruch dieser Anmeldung nicht in einer früheren amerikanischen Patentanmeldung laut dem ersten Paragraphen des Absatzes 35 der Zivilprozeßordnung der Vereinigten Staaten, Paragraph 122 offenbart ist, erkenne ich gemäss Absatz 37, Bundesgesetzbuch, Paragraph 1.56(a) meine Pflicht zur Offenbarung von Informationen an, die zwischen dem Anmeldedatum der früheren Anmeldung und dem nationalen oder PCT internationalen Anmeldedatum dieser Anmeldung bekannt geworden sind.

I hereby claim the benefit under Title 35. United States Code. §120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, §122, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, §1.56(a) which occured between the filing date of the prior application and the national or PCT international filing date of this application.

PCT/DE00/00077	11.01.2000	anhängig	pending
(Application Serial No.) (Anmeldeseriennummer)	(Filing Date D, M, Y) (Anmeldedatum T, M, J)	(Status) (patentiert, anhängig, aufgegeben)	(Status) (patented, pending, abandoned)
(Application Serial No.) (Anmeldeseriennummer)	(Filing Date D,M,Y) (Anmeldedatum T, M; J)	(Status) (patentiert, anhängig, aufgeben)	(Status) (patented, pending, abandoned)

Ich erkläre hiermit, dass alle von mir in der vorliegenden Erklärung gemachten Angaben nach meinem besten Wissen und Gewissen der vollen Wahrheit entsprechen, und dass ich diese eidesstattliche Erklärung in Kenntnis dessen abgebe, dass wissentlich und vorsätzlich falsche Angaben gemäss Paragraph 1001, Absatz 18 der Zivilprozessordnung der Vereinigten Staaten von Amerika mit Geldstrafe belegt und/oder Gefängnis bestraft werden koennen, und dass derartig wissentlich und vorsätzlich falsche Angaben die Gültigkeit der vorliegenden Patentanmeldung oder eines darauf erteilten Patentes gefährden können.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true, and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Form PTO-FB-240 (8-83)
Patent and Trademark Office-U.S. DEPARTMENT OF COMMERCE

Holby M. Abern (P47,372), Robert M. Barrett (30,142), Alan L. Barry (30,819), Thomas C. Basso (46,541), Jeffrey H. Canfield (38,404), Robert W. Connors (46,639), Amy J. Gast (41,773), Timothy L. Harney (38,174), Patricia A. Kane (46,446), Michael S. Leonard (37,557), Edward A. Lehman (22,312), Adam H. Masia (35,602), Dante J. Picciano (33,543), Renato L. Smith (45,117), Maurice E. Teixeira (45,646), William E. Vaughan (39,056), Austin Victor (47,154), and all members of the firm of Bell, Boyd & Lloyd LLC.

17

German Language Declaration

VERTRETUNGSVOLLMACHT: Als benannter Erfinder beauftrage ich hiermit den nachstehend benannten Patentanwalt (oder die nachstehend benannten Patentanwälte) und/oder Patent-Agenten mit der Verfolgung der vorliegenden Patentanmeldung sowie mit der Abwicklung aller damit verbundenen Geschäfte vor dem Patent- und Warenzeichenamt: *(Name und Registrationsnummer anführen)*

POWER OF ATTORNEY: As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith. *(list name and registration number)*

(SEE ATTACHED SHEET)

And I hereby appoint

Customer No.

Telefongespräche bitte richten an: *(Name und Telefonnummer)*

Direct Telephone Calls to: *(name and telephone number)*

Ext. ____________

Postanschrift:

Send Correspondence to:

Bell, Boyd & Lloyd LLC
Three First National Plaza, 70 West Madison Street, Suite 3300 60602-4207 Chicago, Illinois
Telephone: (001) 312 372 11 21 and Facsimile (001) 312 372 20 98
or
Customer No.

Voller Name des einzigen oder ursprünglichen Erfinders:	Full name of sole or first inventor:
Dr. GERHARD SPITZ	Dr. GERHARD SPITZ
Unterschrift des Erfinders / Datum: Gerhard Spitz 23.7.2001	Inventor's signature / Date
Wohnsitz: MUENCHEN, DEUTSCHLAND DEX	Residence: MUENCHEN, GERMANY
Staatsangehörigkeit: DE	Citizenship: DE
Postanschrift: ST.-CAJETAN-STR. 13	Post Office Addess: ST.-CAJETAN-STR. 13
81669 MUENCHEN	81669 MUENCHEN
Voller Name des zweiten Miterfinders (falls zutreffend):	Full name of second joint inventor, if any:
Unterschrift des Erfinders / Datum	Second Inventor's signature / Date
Wohnsitz: ,	Residence: ,
Staatsangehörigkeit	Citizenship
Postanschrift	Post Office Address

(Bitte entsprechende Informationen und Unterschriften im Falle von dritten und weiteren Miterfindern angeben).

(Supply similar information and signature for third and subsequent joint inventors).

Form PTO-FB-240 (8-83)
Patent and Trademark Office-U.S. Department of COMMERCE